A Mortar BDD method for solving flow in stochastic discrete fracture networks

Géraldine Pichot, Baptiste Poirriez, Jocelyne Erhel, Jean-Raynald de Dreuzy

A Mortar BDD method for solving flow in stochastic discrete fracture networks

Géraldine Pichot[1], Baptiste Poirriez[2], Jocelyne Erhel[1], and Jean-Raynald de Dreuzy[3]

1 Introduction

In geological media, the large variety and complex configurations of fractured networks make it difficult to describe them precisely. A relevant approach is to model them as Discrete Fracture Networks (DFN)[10, 19], with statistical properties in agreement with in situ experiments [15, 13, 14]. A DFN is a 3D domain made of 2D fractures intersecting each other. Steady state flow in DFN is considered, the rock matrix is assumed impervious. Following a Monte-Carlo approach, a large number of DFN has to be generated and for each, a flow problem has to be solved whatever the complexity of the generated networks. Moreover time and memory costs for each simulation should be as lower as possible.

A nonconforming discretization of DFN allows to reduce the number of unknowns and facilitate mesh refinement. Sharp angles are managed by a staircase-like discretizations of the fractures' contours [34]. The non-matching feature at the fractures' intersections is handled via a Mortar method [4, 5, 1] developed for DFN in [33, 34] for a mixed hybrid finite element formulation. It consists in defining, for each intersection between fractures, master and slave sides. Due to the staircase-like discretizations, a shared edge may be labeled several times with master and/or slave properties, it is called in the paper a multi-labeled edge. Continuity conditions are enforced between the unknowns on both sides. The derived linear system has only inner and master traces of hydraulic head as unknowns. The matrix A of this system is a symmetric definite positive (SPD) arrow matrix in presence of Dirichlet boundary conditions [34].

The challenge is to solve such linear systems with millions of unknowns [17]. Direct solvers (like Cholmod [11]) are very efficient for small systems but suffer from a high need of RAM memory when the system size becomes too large. Among iterative solvers, multigrid methods are very efficient for most networks but for some, the convergence rate is very slow [35, 17]. Preconditioned Conjugate Gradient (PCG) is efficient and robust for every network tested [35]. The natural decomposition of the matrix A in subdomains encourages the use of domain decomposition methods [7, 36, 31, 24]. The Schur complement of the matrix A is SPD and yields an interface system with only master unknowns. This interface system can be solved iteratively

[1] INRIA Rennes Bretagne-Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: {geraldine.pichot}{jocelyne.erhel}@inria.fr .[2] IRISA, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: baptiste.poirriez@irisa.fr .[3] Géosciences Rennes, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: Jean-Raynald.de-Dreuzy@univ-rennes1.fr

with PCG. The unknowns on inner edges are then derived locally in each fracture plane by solving small local linear systems, with a direct solver for example.

Among possible preconditioners, the balancing domain decomposition (BDD) method is based on a Neumann-Neumann preconditioner coupled with a coarse level solver, to improve the preconditioner as the number of subdomains increases [29, 30, 27]. BDD method applied to mixed finite element is done in [12]. The application to a nonconforming discretization is proposed in [18, 32]. Meanwhile, an alternative method has been developed, the Balancing Domain Decomposition by Constraints (BDDC) [16], later applied to mortar discretization for geometrically nonconforming partitions in [26].

In this paper, we use the BDD algorithm proposed in [32, 35] to solve the linear system arising from a nonconforming discretization of DFN. The coarse level is defined following [37] and balancing is implemented as a preconditioning matrix [21]. The algorithm is implemented in C++ in the parallel software SIDNUR [35]. For DFN, choosing one subdomain given by one fracture, instead of a set of fractures has shown to be the most time saving decomposition [35].

The paper is organized in four sections. Section 2 describes the flow model. Section 3 recalls the linear system derived from a nonconforming discretization of the DFN. Section 4 is the main contribution of this paper and presents the decomposition in local matrices. We apply the BDD method proposed in [32, 35] for networks satisfying some hypotheses on the mesh. The last section illustrates the application of the solver SIDNUR [35] on three stochastically generated DFN.

2 Flow model

We consider flow in DFN assuming the rock matrix is impervious. In the entire paper, an intersection is uniquely defined as the segment shared by two fractures. We denote Σ_k the kth intersection, $k = 1, ..., N_i$.

Poiseuille's law and mass conservation apply in each fracture plane, denoted Ω_f, $f = 1, ..., N_f$. We assume there is no longitudinal flux at the fracture intersections.

The DFN is embedded in a cube of size L. Some fractures are truncated by the cube faces. Classical permeameter boundary conditions apply on the cube faces. The two opposite faces of the cube with Dirichlet boundary conditions (prescribed value p^D) are called Γ_D ($\Gamma_D \neq \emptyset$) and the lateral faces with homogeneous Neumann boundary conditions are called Γ_N. The boundary of the fracture f is called Γ_f. In the following, we assume there is only one cluster of fractures connected to the Dirichlet boundary conditions and we consider only this cluster.

In each fracture plane, with $x \in \mathbb{R}^2$, the following equations link the unknown hydraulic head scalar function $p(x)$ and the flux per unit length function $u(x)$:

$$\nabla \cdot u(x) = f(x) \quad \text{for } x \in \Omega_f, \tag{1}$$

$$u(x) = -\mathscr{T}(x)\nabla p(x) \quad \text{for } x \in \Omega_f, \tag{2}$$

$$p(x) = p^D(x) \quad \text{on } \Gamma_D \cap \Gamma_f, \tag{3}$$

$$u(x).\nu = 0 \quad \text{on } \Gamma_N \cap \Gamma_f, \tag{4}$$
$$u(x).\mu = 0 \quad \text{on } \Gamma_f \backslash \{(\Gamma_f \cap \Gamma_D) \cup (\Gamma_f \cap \Gamma_N)\}, \tag{5}$$

where ν (respectively μ) denotes the outward normal unit vector of the borders with respect to the fracture Ω_f. The parameter $\mathscr{T}(x)$ is a given SPD transmissivity field (unit $[\mathrm{m}^2.\mathrm{s}^{-1}]$). The function $f(x) \in L^2(\Omega_f)$ represents the sources/sinks.

Let $\mathscr{I}_l$ be a segment shared by several incident fractures, $l = 1,...,N_l$. It can be the intersection itself or only a part of it if intersections overlap. Let F_l be the set of fractures which contains $\mathscr{I}_l$. On each segment, continuity conditions are imposed to ensure the continuity of hydraulic heads and the conservation of fluxes [20], [38]:

$$p_{f,l} = p_l \quad \text{on } \mathscr{I}_l, \forall f \in F_l, \tag{6}$$
$$\sum_{f \in F_l} u_{f,l}.n_{f,l} = 0 \quad \text{on } \mathscr{I}_l, \tag{7}$$

where $p_{f,l}$ is the trace of hydraulic head on $\mathscr{I}_l$ in the fracture Ω_f, p_k is the unknown hydraulic head on the segment $\mathscr{I}_l$ and $u_{f,l}.n_{f,l}$ is the normal flux through $\mathscr{I}_l$ coming from the fracture Ω_f, with $n_{f,l}$ the outward normal unit vector of the segment $\mathscr{I}_l$ with respect to the fracture Ω_f.

3 A Mortar method applied to DFN

3.1 Mesh generation

With a stochastic generation, fractures can cross in a very intricate way. We define the contour of a fracture f as its border and all segments $\mathscr{I}_l$ which belong to f. To preserve a good mesh quality whatever the generated fractured networks, staircase like discretizations of the contour are performed in each fracture plane.

Each fracture is meshed with its own mesh step:

1. A temporary uniform grid is built that encompasses the fracture, with a grid step chosen as input;
2. 1D staircase-like meshes of the contour are built using the centers of the grid elements as discretization points;
3. From these 1D discretizations, a 2D triangle mesh of the fracture is built.

We call shared edges the edges of the triangles that discretize the segments $\mathscr{I}_l$, $l = 1,...,N_l$ within the different fractures in F_l. All other edges are called inner edges. Notice a given segment $\mathscr{I}_l$ may have different discretizations in the different fractures in F_l as shown on figure 1. The total mesh is made of N_{in} inner edges and of N_Σ shared edges. In the following, we will use the subscript *in* to refer to the inner edges and Σ to shared edges.

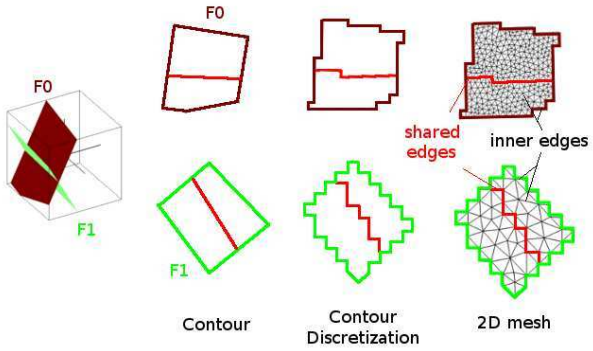

Fig. 1 Mesh generation - Simple example with two fractures

3.2 Derivation of the linear system

The Mortar method applied to DFN is presented in [34]. It consists, for each intersection Σ_k, of choosing a master fracture m and a slave fracture s. We denote $N_m = \sum_{k=1}^{N_i} N_{k,m}$, $N_s = \sum_{k=1}^{N_i} N_{k,s}$, with $N_{k,\{m,s\}}$ the number of edges that discretize the master (respectively slave) side of the intersection Σ_k.

The traces of hydraulic head unknowns are Λ_{in} on inner edges, Λ_m and Λ_s on master and slave edges. Additionnally, each shared edge has an unknown called Λ_Σ. The additional unknowns Λ_Σ allow to deal with multi-labeled edges which belong to several intersections. The unknowns Λ_s and Λ_Σ are derived from Λ_m following the relations (see [34]):

$$\Lambda_s = C\Lambda_m, \tag{8}$$

$$\Lambda_\Sigma = P_m\Lambda_m + P_s\Lambda_s = (P_m + P_sC)\Lambda_m. \tag{9}$$

The matrix C is an intersection block matrix of dimension N_sxN_m, with the block C_k a matrix of size $N_{k,s}$x$N_{k,m}$ for the intersection Σ_k that represents the L^2-projection from the master side to the slave side.

Let denote m_E (respectively s_E) the number of times a shared edge E is labeled with a master (respectively slave) property. Let $n_E = s_E + m_E$. The values (i, j) of the matrices P_m (respectively P_s) of size N_ΣxN_m (respectively N_ΣxN_s) is $\frac{1}{n_E}$ if the unknown $\Lambda_m(j)$ (respectively $\Lambda_s(j)$) is associated to an edge with $\Lambda_\Sigma(i)$ as shared unknown, and 0 otherwise.

At the network scale, the linear system reduces to a system with unknowns Λ_{in} and Λ_m [34]:

$$A\begin{pmatrix}\Lambda_{in}\\ \Lambda_m\end{pmatrix}=\begin{pmatrix}F_{in}\\ F_m\end{pmatrix}. \tag{10}$$

The second member is a vector of dimension $N_{in}+N_m$, which corresponds to the source/sink function, to the imposed Dirichlet and Neumann boundary conditions.

The matrix A is SPD in presence of Dirichlet boundary conditions [34] and writes as:

$$\begin{cases} A &= \begin{pmatrix}A_{in,in} & A_{in,m}\\ A_{in,m}^T & A_{m,m}\end{pmatrix},\\ A_{in,m} &= A_{in,\Sigma}(P_m+P_sC),\\ A_{m,m} &= (P_m+P_sC)^T A_{\Sigma,\Sigma}(P_m+P_sC).\end{cases} \tag{11}$$

The matrix $A_{in,in}$ is a block diagonal matrix of order N_{in} made of blocks $A_{f,in,in}$ associated to the inner edges in the fracture Ω_f.

4 A Mortar BDD method for DFN system

The arrow shape of the matrix A allows to reduce the linear system (10) to an interface problem with only Λ_m as unknowns:

$$S\Lambda_m = B_m, \tag{12}$$

$$S = A_{m,m} - A_{in,m}^T A_{in,in}^{-1} A_{in,m}, \tag{13}$$

$$B_m = F_m - (P_m^T + C^T P_s^T) A_{in,\Sigma}^T A_{in,in}^{-1} F_{in}. \tag{14}$$

with S the Schur complement of size N_mxN_m.

Since S is SPD, the linear system (12) can be solved iteratively via a PCG method. To apply a balancing preconditioner, we need the local Schur complements S_f, $f=1,...,N_f$.

4.1 Local Schur complements

Let $N_{f,m}$ (respectively $N_{f,s}$) be the number of master (respectively slave) unknowns associated with master (respectively slave) edges in the fracture f. Let $N_{f,o}$ be the number of master unknowns associated with the slave edges in the fracture f following the relations (8). Let $N_{f,\Sigma}$ be the number of shared edges in the fracture f. We define the local matrices $(P_m+P_sC)_f$ as:

$$(P_m+P_sC)_f = \begin{pmatrix} P_{f,m} & P_{f,s}C_f \end{pmatrix} \tag{15}$$

with $P_{f,m}$ of size $N_{f,\Sigma}$x$N_{f,m}$ and $P_{f,s}$ of size $N_{f,\Sigma}$x$N_{f,s}$. The matrix C_f of size $N_{f,s}$x$N_{f,o}$ is a block matrix whose blocks C_k are extracted from the matrix C for the intersections Σ_k in the fracture f.

The local problem in the fracture f writes as:

$$A_{f,\Sigma} = \begin{pmatrix} A_{f,in,in} & A_{f,in,\Sigma} \\ A_{f,in,\Sigma}^T & A_{f,\Sigma,\Sigma} \end{pmatrix} \tag{16}$$

Its associated Schur complement writes as: $S_{f,\Sigma} = A_{f,\Sigma,\Sigma} - A_{f,in,\Sigma}^T A_{f,in,in}^{-1} A_{f,in,\Sigma}$.

At the fracture scale, local matrices A_f, of order $(N_{f,in} + N_{f,m} + N_{f,o})$ are built from $A_{f,\Sigma}$:

$$\begin{cases} A_f &= \begin{pmatrix} A_{f,in,in} & A_{f,in,m} \\ A_{f,in,m}^T & A_{f,m,m} \end{pmatrix}, \\ A_{f,in,m} &= \begin{pmatrix} A_{f,in,\Sigma} P_{f,m} & A_{f,in,\Sigma} P_{f,s} C_f \end{pmatrix}, \\ A_{f,m,m} &= \begin{pmatrix} P_{f,m}^T A_{\Sigma,\Sigma} P_{f,m} & P_{f,m}^T A_{\Sigma,\Sigma} P_{f,s} C_f \\ (P_{f,m}^T A_{\Sigma,\Sigma} P_{f,s} C_f)^T & (P_{f,s} C_f)^T A_{\Sigma,\Sigma} P_{f,s} C_f \end{pmatrix}. \end{cases} \tag{17}$$

The block $A_{f,in,m}$ is of size $N_{f,in}$x$(N_{f,m} + N_{f,o})$ and the block $A_{f,m,m}$ is of size $(N_{f,m} + N_{f,o})$x$(N_{f,m} + N_{f,o})$.

The local Schur complement S_f associated to the matrix A_f (17) of the fracture Ω_f writes:

$$S_f = A_{f,mm} - A_{f,in,m}^T A_{f,in,in}^{-1} A_{f,in,m} = (P_m + P_s C)_f^T S_{f,\Sigma} (P_m + P_s C)_f. \tag{18}$$

As each intersection involves two fractures, one slave and one master, the Schur complement S of size N_mxN_m is the sum of the local Schur complements:

$$S = \sum_{f=1}^{N_f} R_f^T S_f R_f, \tag{19}$$

where R_f is the rectriction matrix from the network to the fracture f.

4.2 Neumann-Neumann preconditioner

In the following, a subdomain Ω_f is said to be floating if it does not contain any Dirichlet boundary conditions, non floating otherwise.

The Neumann-Neumann preconditioner [25, 9, 28] writes as:

$$M_{NN}^{-1} = D \sum_f R_f^T S_f^\dagger R_f D, \tag{20}$$

where

$$S_f^\dagger = \begin{cases} S_f^{-1} \text{ if } S_f \text{ is non singular,} \\ \tilde{S}_f^{-1} \text{ otherwise, with } \tilde{S}_f \text{ a non singular approximation of } S_f. \end{cases} \tag{21}$$

The matrix D is a diagonal matrix of order N_m. With a nonconforming discretization, a definition of one fracture as one subdomain and an homogeneous transmissivity, $D = \frac{1}{2}Id$ since each master unknown is defined for an intersection between two subdomains.

From the definition of M_{NN}^{-1}, one needs to solve local subdomain problems with the matrix S_f, like $S_f z_f = r_f$. However the kernel of S_f may not be trivial. If the matrix $(P_m + P_s C)_f$ is of full rank, the kernel of S_f is that if $S_{f,\Sigma}$: $\{0\}$ for a non floating subdomain, else $\{const\}$. We assume that $(P_m + P_s C)_f$ is of full rank if the following conditions are satisfied:

(H1) the master side of an intersection must have the smallest number of discretization edges: $N_{k,m} \leq N_{k,s}, \forall k \in 1, ..., N_i$;

(H2) There are no multi-labeled edges: $n_E = 1$ for each shared edge E yielding: $N_\Sigma = N_m + N_s$.

If the subdomain is floating, in order to get a SPD approximation $\tilde{S}_f$, we add one arbitrary Dirichlet condition, since the kernel is of dimension 1 [35].

4.3 Balancing preconditioner

As the number of subdomains increases, the efficiency of the Neumann-Neumann preconditioner decreases [27] and one has to couple it with a coarse level solver [29, 30]. We use the following balancing preconditioner:

$$M_b^{-1} = P^T M_{NN}^{-1}, \tag{22}$$

as in [37, 21, 35] where the projection matrix P, of order N_m, is defined as:

$$P = I - SZS_c^{-1}Z^T. \tag{23}$$

The matrix Z is a N_mx N_c subspace matrix with full rank, $N_c < N_m$, and $S_c = Z^T SZ$ is the invertible matrix corresponding to the coarse problem.

This formulation is based on the PCG initial value:

$$\Lambda_{m,0} = ZS_c^{-1}Z^T B_m, \tag{24}$$

such that, for all iterations it of PCG, the residuals $r_{it} = S\Lambda_{m,it} - B_m$ satisfy $Z^T r_{it} = 0$ and $Pr_{it} = r_{it}$ [35]. Thus applying (22) is equivalent to apply $P^T M_{NN}^{-1} P + ZS_c^{-1}Z^T$ [37, 35].

A possible choice for the full rank matrix Z is to use a subdomain deflation as defined in [22, 35]. Here $N_c \leq N_f$ and Z is sparse.

5 Numerical experiments

We present preliminary numerical experiments on three random DFN that satisfy hypotheses $(H1)-(H2)$, generated with the software MP_FRAC of the H2OLab platform `http://h2olab.inria.fr/`. We checked there is only one connected cluster. We build the local matrices A_f and use the software SIDNUR which implements the BDD method [35].

5.1 Geometry and boundary conditions

The position of the fractures is taken as uniform in the domain. Their orientation is uniform and their length follows a power law distribution of exponent 2.7 [8]. We take $p^D = 1m$ on the cube face at $y = L/2$ and $p^D = 20m$ on the cube face at $y = -L/2$. The transmissivity tensor is homogeneous and equal to $\mathscr{T} = T\,Id$, with $T = 8.2e-7$ m^2.s^{-1}. We consider 3 networks:

- $L6\_NF28$: L=6 and N_f=28;
- $L10\_NF18$: L=10 and N_f=18;
- $L10\_NF24$: L=10 and N_f=24.

5.2 Mesh procedure and basic optimization

The nonconforming mesh is generated according to the mesh procedure described in subsection 3.1. With this approach, adaptative mesh refinement can be done at the fracture level [2, 3, 39, 6].

A basic mesh coarsening consists in meshing finely only the fractures that take part significantly in the flow. Let us run a first simulation with a coarse mesh step $2*\Delta$. The output flux for each fracture is computed, as well as the total output flux on the output cubic face. We choose to refine, with a mesh step Δ, the fractures that have an output flux above 5 % of the total output flux. The simulation is performed again on this refined mesh.

In table 1, we compare the mesh obtained with this basic mesh coarsening, so-called coarser mesh, with a mesh where the step is Δ for all fractures, so-called fine mesh. The min and mean of the quality mesh criterion $Q_K \in [0;1]$ is also given, where Q_K is defined for each triangle K as [23]:

$$Q_K = 4\sqrt{3}\frac{S_K}{h_s^2}, \tag{25}$$

with S_K the surface of the triangle K and $h_s = \sqrt{\sum_{i=1}^{3} h_i^2}$, with h_i the length of the edge i of the triangle K. The closer Q_K is to 1, the better the triangle quality is.

Table 1 Comparison between a mesh with step Δ for all fractures and a mesh with step Δ for fractures with an output flux above 5 % of the total output flux and $2*\Delta$ otherwise

Simulation name	Δ	Fine mesh - step Δ			Coarser mesh - step Δ or 2Δ		
		Number of edges	Min(Q_K)	Mean(Q_K)	Number of edges	Min(Q_K)	Mean(Q_K)
$L6\_NF28$	0.05	122306	0.43	0.95	90533	0.23	0.95
$L10\_NF18$	0.1	62409	0.45	0.95	57462	0.19	0.95
$L10\_NF24$	0.1	78652	0.51	0.95	67765	0.25	0.95

Table 1 shows that this basic mesh coarsening reduces the number of edges from 7.93 % to 25.96 % at the price of somehow lower mesh quality. Indeed the length of some fractures is too small compared with 2Δ, yielding too few discretization points. As future work, we could define a minimal mesh step per fracture according to its length.

5.3 Solution with SIDNUR

Using the coarser mesh, we solve the linear system (12) with the BDD method. We checked these networks satisfy hypotheses $(H1)-(H2)$. From the computed values of Λ_m, we derive the unknowns Λ_s and Λ_Σ according to $(8)-(9)$. The inner unknowns Λ_{in} are derived locally in each fracture plane by solving small linear systems (see (10)). From these traces of hydraulic head unknowns, one can derive the mean head values and the fluxes [34]. Figures 2, 3 and 4 give the mean head values on the three DFN. Figure 5 displays the mean head values for the DFN $L10\_NF24$ obtained by solving the linear system (12) with CHOLMOD to illustrate the good agreement of the results obtained with the two methods.

Table 2 gives the numbers N_{in}, N_m and N_s with $N_\Sigma = N_m + N_s$ (hypothesis $(H2)$). This table also provides the number of PCG iterations, the final L^2-norm of the residual and the L^2-norm of the relative difference between the solutions $\begin{pmatrix}\Lambda_{in}\\ \Lambda_m\end{pmatrix}$ computed with SIDNUR and with the direct solver CHOLMOD [11].

Table 2 Solution with SIDNUR. Comparison with CHOLMOD

Simulation name	N_{in}	N_m	N_s	# PCG it.	PCG final residual	Comparison with CHOLMOD
$L6\_NF28$	89732	365	436	13	6.02e-17	4.15e-12
$L10\_NF18$	56939	247	276	15	2.47e-18	9.56e-13
$L10\_NF24$	66899	412	454	18	8.71e-19	1.47e-12

On such small linear systems with very small CPU times, the solver SIDNUR is not competitive with respect to a direct solver. However this preliminary test phase demonstrates the possibility of solving linear system arising from a nonconforming discretization of networks satisfying hypotheses $(H1)-(H2)$ with the BDD

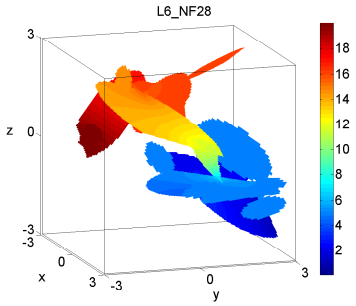

Fig. 2 $L6\_NF28$ - Mean head - SIDNUR

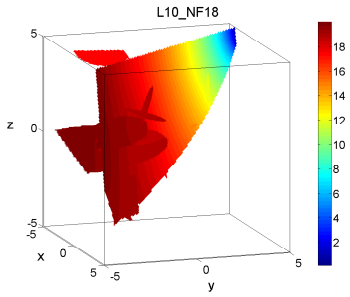

Fig. 3 $L10\_NF18$ - Mean head - SIDNUR

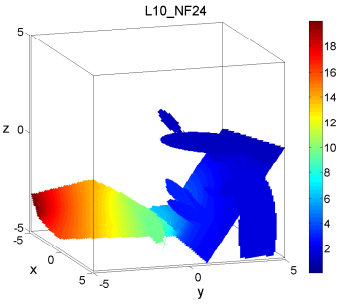

Fig. 4 $L10\_NF24$ - Mean head - SIDNUR

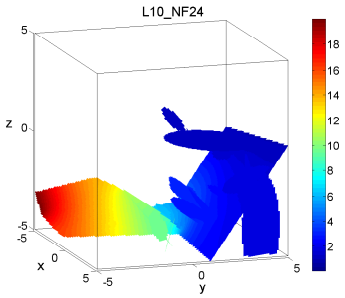

Fig. 5 $L10\_NF24$ - Mean head - CHOLMOD

method. Using SIDNUR relies on a suitable decomposition of the local matrices. Moreover SIDNUR requires less RAM memory than a direct solver and is parallel.

6 Conclusion

This paper describes a Balancing Domain Decomposition method, implemented in the so-called SIDNUR solver, to simulate flow in DFN with a nonconforming mesh. DFN and local matrices are generated with the so-called MP_FRAC software. Our current work is to extend the method to more general discretizations, which do not satisfy hypotheses $(H1) - (H2)$, in the perspective of solving linear systems with several millions of unknowns. The parallelism of SIDNUR will be very helpful to reduce the time and memory costs. Moreover the very basic technic we use to coarsen the mesh could be improved by defining suitable *a posteriori* estimators.

Acknowledgements This work was supported by the French National Research Agency, with the ANR-07-CIS7 project MICAS, and by INRIA with the ARC-INRIA GEOFRAC project.

References

1. Arbogast, T., Cowsar, L.C., Wheeler, M.F., Yotov, I.: Mixed finite element methods on nonmatching multiblock grids. SIAM J. Numer. Anal **37**, 1295–1315 (2000)
2. Arnold, D.N., Brezzi, F.: Mixed and nonconforming finite element methods : implementation, postprocessing and error estimates. ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique **19**(1), 7–32 (1985). URL `http://eudml.org/doc/193443`
3. Bernardi, C., Hecht, F.: Error indicators for the mortar finite element discretization of the Laplace equation. Mathematics of Computation **71**(240), 1371–1403 (2002)
4. Bernardi, C., Maday, Y., Patera, A.T.: Domain decomposition by the mortar element method. In: Asymptotic and numerical methods for partial differential equations with critical parameters (Beaune, 1992), *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, vol. 384, pp. 269–286. Kluwer Acad. Publ., Dordrecht (1993)
5. Bernardi, C., Maday, Y., Patera, A.T.: A new nonconforming approach to domain decomposition: the mortar element method. In: Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991), *Pitman Res. Notes Math. Ser.*, vol. 299, pp. 13–51. Longman Sci. Tech., Harlow (1994)
6. Bernardi, C., Rebollo, T.C., Hecht, F., Mghazli, Z.: Mortar finite element discretization of a model coupling darcy and stokes equations. ESAIM: Mathematical Modelling and Numerical Analysis **42**(3), 375–410 (2008). URL `http://eudml.org/doc/250402`
7. Bjørstad, P.E., Widlund, O.B.: Solving elliptic problems on regions partitioned into substructures. In: G. Birkhoff, A. Schoenstadt (eds.) Elliptic Problem Solvers II, pp. 245–256. Academic Press, New York (1984)
8. Bonnet, E., Bour, O., Odling, N., Davy, P., Main, I., Cowie, P., Berkowitz, B.: Scaling of fracture systems in geological media. Reviews of Geophysics **39**(3), 347–383 (2001)
9. Bourgat, J.F., Glowinski, R., Le Tallec, P., Vidrascu, M.: Variational formulation and algorithm for trace operator in domain decomposition calculations. In: T. Chan, R. Glowinski, J. Périaux, O. Widlund (eds.) Domain Decomposition Methods, pp. 3–16. SIAM, Philadelphia, PA (1989)
10. Cacas, M.C., Ledoux, E., de Marsily, G., Barbeau, A., Calmels, P., Gaillard, B., Magritta, R.: Modeling fracture flow with a stochastic discrete fracture network: calibration and validation. 1. the flow model. Water Resources Research **26**(3), 479–489 (1990)
11. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans. Math. Softw. **35**(3), 22:1–22:14 (2008). DOI 10.1145/1391989.1391995. URL `http://doi.acm.org/10.1145/1391989.1391995`
12. Cowsar, L.C., Mandel, J., Wheeler, M.F.: Balancing domain decomposition for mixed finite elements. Mathematics of computation **64**(211), 989–1015 (1995). DOI {10.2307/2153480}
13. Davy, P., Bour, O., de Dreuzy, J.R., Darcel, C.: Flow in multiscale fractal fracture networks. In: 261 (ed.) Geological society, London, special publications, pp. 31–45 (2006)
14. Davy, P., Le Goc, R., Darcel, C., Bour, O., de Dreuzy, J.R., Munier, R.: A likely universal model of fracture scaling and its consequence for crustal hydromechanics. Journal of Geophysical Research **115**, B10,411 (2010)
15. Dershowitz, W.S., Einstein, H.H.: Characterizing rock joint geometry with joint system models. Rock Mechanics and Rock Engineering **21(1)**, 2151 (1988)
16. Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25**, 246–258 (2003)
17. de Dreuzy, J.R., Pichot, G., Poirriez, B., Erhel, J.: Synthetic benchmark for modeling flow in 3D fractured media. Computers & Geosciences **50**, 59–71 (2013). DOI 10.1016/j.cageo.2012.07.025. URL `http://hal.inria.fr/hal-00735675`
18. Dryja, M., Proskurowski, W.: On preconditioners for mortar discretization of elliptic problems. Numerical Linear Algebra with Applications **10**(1-2), 65–82 (2003). DOI 10.1002/nla.312. URL `http://dx.doi.org/10.1002/nla.312`

19. Dverstop, B., Andersson, J.: Application of the discrete fracture network concept with field data: possibilities of model calibration and validation. Water Resources Research **25(3)**, 540550 (1989)
20. Erhel, J., de Dreuzy, J.R., Poirriez, B.: Flow simulations in three-dimensional discrete fracture networks. SIAM Journal on Scientific Computing **31**(4), 2688–2705 (2009). DOI DOI:10.1137/080729244
21. Erhel, J., Guyomarc'h, F.: An augmented conjugate gradient method for solving consecutive symmetric positive definite systems. SIAM Journal on Matrix Analysis and Applications **21**(4), 1279–1299 (2000)
22. Frank, J., Vuik, C.: On the construction of deflation-based preconditioners. SIAM J. Sci. Comput. **23**(2), 442–462 (2001). DOI 10.1137/S1064827500373231. URL http://dx.doi.org/10.1137/S1064827500373231
23. Frey, P.J., George, P.L.: Maillages : applications aux éléments finis. Hermès sciences publ. DL1999 (53-Mayenne), Paris. URL http://opac.inria.fr/record=b1094298
24. Gander, M.J., Tu, X.: On the origins of iterative substructuring methods. In: Domain Decomposition Methods (2013)
25. Glowinski, R., Wheeler, M.F.: Domain decomposition and mixed finite element methods for elliptic problems. In: R. Glowinski, G.H. Golub, G.A. Meurant, J.P. Eds (eds.) Domain Decomposition Methods for Partial Differential Equations, pp. 144–172. SIAM, Philadelphia (1988)
26. Kim, H.H., Dryja, M., Widlund, O.B.: A bddc method for mortar discretizations using a transformation of basis. SIAM J. Numerical Analysis **47**(1), 136–157 (2008)
27. Le Tallec, P.: Domain decomposition methods in computational mechanics. In: J.T. Oden (ed.) Computational Mechanics Advances, vol. 1 (2), pp. 121–220. North-Holland (1994)
28. Le Tallec, P., De Roeck, Y., Vidrascu, M.: Domain decomposition methods for large linearly elliptic three-dimensional problems. Journal of Computational and Applied Mathematics **34**, 93–117 (1991)
29. Mandel, J.: Balancing domain decomposition. Communications in Applied Numerical Methods **9**, 233241 (1993)
30. Mandel, J., Brezina, M.: Balancing domain decomposition: Theory and computations in two and three dimensions. Tech. Rep. UCD/CCM 2, Center for Computational Mathematics, University of Colorado at Denver (1993)
31. Meurant, G.: Computer Solution of Large Linear Systems. Elsevier Science B.V. (1999)
32. Pencheva, G., Yotov, I.: Balancing domain decomposition for mortar mixed finite element methods. Numerical Linear Algebra with Applications **10**(1-2), 159–180 (2003). DOI 10.1002/nla.316. URL http://dx.doi.org/10.1002/nla.316
33. Pichot, G., Erhel, J., de Dreuzy, J.R.: A mixed hybrid mortar method for solving flow in discrete fracture networks. Applicable Analysis **89**(10), 1629–1643 (2010)
34. Pichot, G., Erhel, J., de Dreuzy, J.R.: A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks. SIAM Journal on Scientific Computing **34(1)**, B86B105. (20 pages) (2012)
35. Poirriez, B.: Etude et mise en oeuvre d'une méthode de sous-domaines pour la modélisation de l'écoulement dans des réseaux de fractures en 3d. Ph.D. thesis, University of Rennes 1 (2011)
36. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford Science Publications (1999)
37. Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A.: Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. J. Sci. Comput. **39**(3), 340–370 (2009). DOI 10.1007/s10915-009-9272-6. URL http://dx.doi.org/10.1007/s10915-009-9272-6
38. Vohralik, M., Maryska, J., Severyn, O.: Mixed and nonconforming finite element methods on a system of polygons. Applied Numerical Mathematics **57**, 176–193 (2007)
39. Wheeler, M.F., Yotov, I.: A posteriori error estimates for the mortar mixed finite element method. SIAM J. Numer. Anal. **43**(3), 1021–1042 (2005). DOI 10.1137/S0036142903431687. URL http://dx.doi.org/10.1137/S0036142903431687